# 

# Book Proposal

Proposed Book Title: CockroachDB: The Definitive Guide

Subtitle: Distributed Data at Scale

Author(s): Jesse Seldess, Ben Darnell, 3rd author TBD

Author title(s) and affiliation(s):

* VP of Education at Cockroach Labs (Jesse)
* Co-Founder & Chief Architect at Cockroach Labs (Ben)

Mailing address(es):

Phone number:

Preferred Email address(es):

## About the Author(s)

*Author biography (write in the 3rd person as you would like the bio to appear on Amazon):*

Please be sure the biography answers the question:

*Why are you the best person to write this book?*

**Jesse Seldess**

Jesse Seldess is the VP of Education at Cockroach Labs, where he leads the documentation and training teams. He has nearly 20 years of experience in technical documentation, and has built teams from the ground up at Cockroach Labs and AppNexus (now Xander). Outside of work, Jesse is the author of several collections of poetry, most recently SEVERAL ROTATIONS (Kenning Editions, 2019), which stores its code-generated final section in CockroachDB. He lives in Brooklyn, NY with his wife and two children.

LinkedIn profile: <https://www.linkedin.com/in/jesseseldess/>

Twitter handle: n/a

Author public speaking samples (YouTube, etc.): [Write the Docs](https://www.youtube.com/watch?v=YLXNJXQyYoE), [Cockroach Labs Live](https://youtu.be/z74pvLFyKKg)

**Ben Darnell**

Ben Darnell is the co-founder and Chief Architect at Cockroach Lab, where he built the distributed consensus protocols that underpin CockroachDB’s transactional model. He started his career at Google and then went on to a series of startups where he saw firsthand the need for better scalable storage systems. When he’s not working he likes to travel the globe in search of food and adventure.

LinkedIn profile: n/a

Twitter handle: n/a

+ Wikipedia: <https://en.wikipedia.org/wiki/Ben_Darnell>

Author public speaking samples (YouTube, etc.): [ACM Applicative](https://www.youtube.com/watch?v=hWNwI5q01gI), [Golang UK](https://www.youtube.com/watch?v=33oqpLmQ3LE), [Software Engineering Daily](https://www.youtube.com/watch?v=tx04GPdzGX8)

## Marketing Description

*In 1-2 paragraphs, summarize what the book is about, like you would pitch it to a potential reader on the back cover. What makes your book unique in the marketplace?*

The first edition of this expert-written guide provides a comprehensive overview of CockroachDB, an elastic, indestructible SQL database built to handle the demands of today's data-driven world. The guide reviews the advantages of building on a distributed SQL database, and teaches readers how to build applications that scale elastically and provide seamless delivery for their end-users, while remaining exceptionally resilient and indestructible.

Authors Jesse Seldess, Ben Darnell, and \_\_\_\_\_ provide practical guidance for application developers and deployment optimizations for DevOps teams. Those familiar with distributed systems will learn the benefits of strong data correctness and consistency guarantees, as well as optimizations for delivering ultra-low latencies to globally distributed end users.

In 17 chapters, readers will learn:

* How to plan and build applications for distributed infrastructure, including data modeling and schema design
* How to migrate data into CockroachDB
* How to read and write data and run ACID transactions across distributed infrastructure
* How to optimize queries for performance across a geographically distributed replicas
* How to plan a CockroachDB deployment for resiliency across single-region and multi-region clusters
* How to secure, monitor, and optimize the CockroachDB deployment

## About the Topic

*Briefly explain the topic and why it is important.*

Databases of today were not built for modern applications. Many predate ecommerce and SaaS, and are unfit to realize the benefits of the cloud. While companies are rapidly evolving, the database has remained largely unchanged -- unfit and unable to keep up with where fast companies need to go. They have become a festering ground for operational complexity -- inflexible, unyielding, and untrustworthy. Millions of dollars are lost due to outages and downtime. App teams suffer through manual drudgery just to keep their applications running.

CockroachDB was written from scratch for the cloud, architected to scale elastically to handle the demands of tomorrow. Cloud-native, open source, and indestructible, CockroachDB makes it easier to build and scale modern applications.

## 

## Audience

*In 1-2 paragraphs, explain who the primary audience is for your book. What professional positions do they hold? What positions are they considering next in their careers? What knowledge do you assume of them? What books can you assume they have read? What skills can you assume they have mastered?*

*Estimate how many people will use this technology. Please state any applicable statistics (e.g., Google Trends, analyst reports, blogs, leading companies adopting the topic of your book) indicating market use or market potential.*

*Please provide some scenarios that indicate how the audience will use your book. For example, will readers refer to it daily as a reference? Will they read it once to learn the concepts and then refer to it occasionally?*

**Audience:**

Innovative Developers: The Definitive Guide is for innovative developers building modern applications. It will teach developers how to build and ship apps with fewer obstacles on a database that just works.

Architects: The Definitive Guide is for modern architects delivering scalable, resilient apps across their IT ecosystem. Architects will learn how to provide low latency and faster performance in distributed applications.

DevOps teams: The Definitive Guide is for strategic IT operators managing applications with data-intensive workloads. Operators will learn how to deploy the effectiveness of a truly elastic database that efficiently distributes data to meet any workload demand, wherever it’s deployed.

**Market Position & Adoption:**

Most databases available simply were not designed to meet the needs of scale-out distributed systems. No matter how much they are modified or extended to address these unique needs, they still struggle.

DistributedSQL has emerged over the past few years to deliver global (or local) transactional guarantees for scale out cloud applications. It delivers a wholly new database architecture distributed at its foundation, delivering easy scale, ultra resilience, and geo-located data.

According to [DB-Engines Ranking](https://db-engines.com/en/ranking/relational+dbms), CockroachDB has trended upwards in popularity over the last year. It’s currently the 35th most popular relational database, a jump from the 45th spot in 2019. While it’s only the [66th most popular database](https://db-engines.com/en/ranking) overall, that's a significant leap from its ranking of 94 in 2019.

CockroachDB has 18.6k stars in [Github](https://github.com/cockroachdb/cockroach) for 2020. According to its [star history](https://star-history.t9t.io/#cockroachdb/cockroach), CockroachDB has experienced significant rating increases every year since its debut in 2015, punctuated by a 3k star jump from 2019 to 2020.

CockroachDB is used by hundreds of organizations worldwide, and in nearly every industry. Customers [include](https://www.cockroachlabs.com/customers) Comcast, Bose, SpaceX, Lush, Nubank, Rubrik, and top-5 US banks.

**Use of the Book:**

Readers of this book should use it first to learn core concepts of distributed SQL systems, then as a regular reference as they build and deploy their own applications.

# What the reader will learn — and how to apply it

By the end of this book the reader will understand:

* The historical context of CockroachDB and distributed SQL
* The architecture and capabilities of CockroachDB
* Suitable use cases for CockroachDB

And the reader will be able to:

* Get started with CockroachDB
* Build effective applications on CockroachDB
* Run a CockroachDB cluster in production

## Keywords

*What are the trending terms for the topics covered in your book?*

distributed database, SQL, distributed SQL, NewSQL, scalability, scale-out, elasticity, resilience, fault tolerance, OLTP, transactions, ACID, serializable

## Other Book Features

*Will there be a github site for code samples?*

Parts of the book may reference code or configuration files hosted on one of Cockroach Lab’s public GitHub repositories.

## Software Dependencies

*What software updates or releases could potentially impact the release of your book?*

By the expected publication date of July 2021, Cockroach Labs will release 2 major versions of CockroachDB (20.2 in October 2020, 21.1 in April 2021), with another major version scheduled 4 months following publication (21.2 in October 2021). The majority of topics in this book should not change based on these releases, but there will be some exceptions. For example, by the time of publication, our geospatial capabilities may be robust enough to warrant coverage, and soon after publication, with the 21.2 release, our multi-region deployment UX may improve significantly.

## Competing Titles

*What print books compete with your book? Please list at least 3 books from other publishers and include title, author, ISBN, publisher name and year. Explain how your book will be different or better in timing, content, coverage, approach or tone than each competing title.*

1. Mastering PostgreSQL 11: Expert Techniques to Build Scalable, Reliable, and Fault-Tolerant Database Applications, 2nd Edition, Hans-Jurgen Schonig, 9781789537819. Packt, 2018 (AMZ rank: +1m) -- Although this book approaches its subject in a similar fashion as our book, it’s focused on an entirely different database.
2. Mastering MongoDB 4.x: Expert Techniques to Run High-Volume and Fault-Tolerant Database Solutions Using MongoDB 4.x, 2nd Edition, Alex Giamas, 9781789617870. Packt, 2019 (AMZ rank: +1m) -- Similar to our book, this title comprehensively covers the major facets of a non-traditional database. However, it’s focus is on a different database.
3. High Availability for the LAMP Stack: Eliminate Single Points of Failure and Increase Uptime for Your Linux, Apache, MySQL, and PHP Based Web Applications, Jason Cannon, 9781505607062, CreateSpace Independent Publishing Platform, 2014 (AMZ rank: +1m) -- This book provides step-by-step guidance on how to address and eliminate single points of failure for web applications built through a number of platforms. Although our book provides similar guidance by teaching readers how to build applications on CockroachDB and troubleshoot issues during the process, its focus is on a single database rather than multiple ones.

## Related O’Reilly Titles

*What O’Reilly book(s) cover similar topics or related technology?*

1. [Designing Distributed Systems](https://learning.oreilly.com/library/view/designing-distributed-systems/9781491983638/), Brendan Burns, 9781491983645, O’Reilly Media, 2018 -- This guide presents a collection of repeatable, generic patterns for developers to help build distributed systems from scratch. Our book takes a deep dive into a specific distributed database, geared not only for developers, but for architects and IT operators as well.
2. [Cassandra: The Definitive Guide, 3rd Edition](https://learning.oreilly.com/library/view/cassandra-the-definitive/9781098115159/), Jeff Carpenter, Eben Hewitt, 9781098115166, O’Reilly Media, 2020 -- This book is similar to our title in that it covers nearly every facet of an open-source database. It differs from our title in its specific coverage of a NoSQL database, as opposed to our book’s focus on a distributed SQL / NewSQL database.
3. [MongoDB: The Definitive Guide, 3rd Edition](https://learning.oreilly.com/library/view/mongodb-the-definitive/9781491954454/), Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, 9781491954461, O’Reilly Media, 2020 -- Similar to our book, this guide provides an in-depth overview of an open-source database. Unlike our book, the guide is focused on a NoSQL database, whereas our book covers a distributed SQL / NewSQL database.

## Book Outline

*Include a detailed outline for the project here, following the example format below. (If proposing a second edition, please highlight changes from the previous edition.)*

*Additional Resources:*

* [*SIGMOD Paper*](https://resources.cockroachlabs.com/scale-resilience/cockroachdb-the-resilient-geo-distributed-sql-database-sigmod-2020) *(*[*alternate link*](https://cdn2.hubspot.net/hubfs/1753393/guides/White%20Paper%20%7C%20CockroachDB%20The%20Resilient%20Geo-Distributed%20SQL%20Database%20(SIGMOD%202020).pdf?__hstc=145681515.2f58e91b372255268815552f35066385.1605641279624.1605641279624.1605641279624.1&__hssc=145681515.2.1605641279625&__hsfp=3156913768)*)*
* [*CockroachDB Internal Message Guide*](https://docs.google.com/document/d/1s3KgctMuhaFAimoCOtg_5HOHu-RqzLCLMcCYDDyUt2E/edit)
* [*Cockroach U*](https://university.cockroachlabs.com/catalog)*:* 
  + *Getting Started*
  + *For Python developers*
* Preface
  + What is CockroachDB?
    - [Message Guide] Section 2: Product Message
  + Is this book for you?
  + What’s in this book?
  + Conventions used in this book
  + Etc.
* Chapter 1: Why CockroachDB?
  + A brief history of databases
    - [Cockroach U] [A Brief History of Databases](https://www.youtube.com/watch?v=PA3LtpwfFwQ&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=3&t=0s)
  + Distributed SQL: The future of databases
    - [Cockroach U] [The Future of Databases: Distributed SQL](https://www.youtube.com/watch?v=eFvFGd_rpXU&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=4&t=0s)
  + CockroachDB capabilities
    - Scale, resilience, consistency, locality
      * [Cockroach U]
        + [CockroachDB: A True Distributed SQL Database](https://www.youtube.com/watch?v=ZAzgd4xvV7o&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=5&t=0s)
        + [Availability and Durability in a 3-node cluster](https://www.youtube.com/watch?v=1IPXUegnwdY&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=12)
        + [Resiliency in the Cluster](https://www.youtube.com/watch?v=_wT9cuc9E2c&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=15)
        + [Fault Tolerance and Automated Recovery](https://www.youtube.com/watch?v=wFRMYfy-RUI&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=17)
        + [ACID Transactions](https://www.youtube.com/watch?v=M6zbQ5_sgq8&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=19)
        + [Locality](https://www.youtube.com/watch?v=3uNIuv-NQcU&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=21)
        + [Serializable Isolation](https://www.youtube.com/watch?v=kHkvYLzc85Y&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=4)
      * [Docs]
        + [Product FAQ](https://www.cockroachlabs.com/docs/stable/frequently-asked-questions.html)
  + CockroachDB performance
    - Predictable throughput and latency at scale, latest benchmarks
      * [Docs]
        + [Performance overview](https://www.cockroachlabs.com/docs/stable/performance.html)
        + [TPC-C benchmarking instructions](https://www.cockroachlabs.com/docs/stable/performance-benchmarking-with-tpc-c-10-warehouses.html)
      * [Blog] [CockroachDB Passes TPC-C 100k](https://www.cockroachlabs.com/blog/tpcc-100k/)
      * [[SIGMOD](https://resources.cockroachlabs.com/scale-resilience/cockroachdb-the-resilient-geo-distributed-sql-database-sigmod-2020)] Section 6: Evaluation
  + Common use cases
    - System of record, Identity access management (IAM), Metadata layer, General database
      * [Message guide] Section 3: Use Cases
* Chapter 2: The CockroachDB Architecture
  + Core concepts
    - Cluster, node, range, replica, leaseholder, raft leader, raft log, gateway
      * [Docs]
        + [Architecture overview](https://www.cockroachlabs.com/docs/stable/architecture/overview.html)
        + [Reads and Writes Overview](https://www.cockroachlabs.com/docs/stable/architecture/reads-and-writes-overview.html)
      * [Cockroach U]
        + [The Keyspace, Replicas, and Ranges](https://www.youtube.com/watch?v=LgbrmIjH0cU&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=10)
        + [The Raft Protocol](https://www.youtube.com/watch?v=k5BR9m8o9ec&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=11)
  + Architecture layers
    - SQL, Transaction, Distribution, Replication, Storage
      * [[Docs] Architecture layer docs](https://www.cockroachlabs.com/docs/stable/architecture/overview.html)
      * [SIGMOD] Section 2, System Overview
  + Example: Life of a distributed transaction
    - [[Docs] Life of a distributed transaction](https://www.cockroachlabs.com/docs/v20.2/architecture/life-of-a-distributed-transaction.html)
* Chapter 3: Getting Started
  + Starting a test cluster
    - Cockroach Cloud free tier (best for dev testing)
      * [[Docs] Cockroach Cloud free cluster](https://www.cockroachlabs.com/docs/cockroachcloud/stable/cockroachcloud-quickstart.html)
    - Local cluster (single-node, multi-node, or in-memory)
      * Installing CockroachDB
        + [[Cockroach U] Installing CockroachDB](https://www.youtube.com/watch?v=s6kXEev3H7g&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=6&t=0s)
        + [[Docs] Installing CockroachDB](https://www.cockroachlabs.com/docs/stable/install-cockroachdb.html)
      * Starting CockroachDB
        + [[Cockroach U]: Spinning up and Scaling Out](https://www.youtube.com/watch?v=dNSINb5MGhI&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=14&t=0s)
        + [Docs]

[Local multi-node (and variations, e.g., secure, kubernetes, docker)](https://www.cockroachlabs.com/docs/stable/start-a-local-cluster.html)

[Local single-node](https://www.cockroachlabs.com/docs/stable/cockroach-start-single-node.html)

[Local in-memory](https://www.cockroachlabs.com/docs/stable/cockroach-demo.html)

* + Connecting to a cluster
    - Using the Built-in SQL shell
      * [[Cockroach U] Connecting with the SQL Shell](https://www.youtube.com/watch?v=3LN2w4ZjiAo&list=PL_QaflmEF2e8Prn7r7CIyBKsHPgsgNO_1&index=8&t=0s)
      * [Docs]
        + [Start a Local Cluster](https://www.cockroachlabs.com/docs/stable/start-a-local-cluster.html)
        + [Learn CockroachDB SQL](https://www.cockroachlabs.com/docs/stable/learn-cockroachdb-sql.html)
        + [cockroach sql](https://www.cockroachlabs.com/docs/stable/cockroach-sql.html)
        + [cockroach demo](https://www.cockroachlabs.com/docs/stable/cockroach-demo.html)
    - Using a driver or ORM
      * Docs
        + [Docs]

[Hello world tutorials](https://www.cockroachlabs.com/docs/stable/hello-world-example-apps.html)

[CockroachCloud Quickstart](https://www.cockroachlabs.com/docs/cockroachcloud/stable/cockroachcloud-quickstart.html)

[Connect to Your CockroachCloud Cluster](https://www.cockroachlabs.com/docs/cockroachcloud/stable/cockroachcloud-connect-to-your-cluster.html)

* + - * + [[Cockroach U] Connecting to CockroachCloud](https://www.youtube.com/watch?v=7pg_bR5zJVc&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q)
  + Exploring database features
    - Docs
      * [Replication and Rebalancing](https://www.cockroachlabs.com/docs/stable/demo-replication-and-rebalancing.html)
      * [Fault Tolerance and Recovery](https://www.cockroachlabs.com/docs/stable/demo-fault-tolerance-and-recovery.html)
      * [Serializable Transactions](https://www.cockroachlabs.com/docs/stable/demo-serializable.html)
      * [JSON Support](https://www.cockroachlabs.com/docs/stable/demo-json-support.html)
      * [Use the Admin UI](https://www.cockroachlabs.com/docs/stable/admin-ui-access-and-navigate.html)
  + Running a Test App
    - [Docs]
      * [Hello world sample apps and tutorials](https://www.cockroachlabs.com/docs/stable/hello-world-example-apps.html)
      * [Roachdata sample apps and tutorials](https://www.cockroachlabs.com/docs/stable/build-a-spring-app-with-cockroachdb-jdbc.html)
* Chapter 4: Schema Design
  + Understanding SQL types and constraints
    - [Docs]
      * [Data types](https://www.cockroachlabs.com/docs/stable/data-types.html)
      * [Constraints](https://www.cockroachlabs.com/docs/stable/constraints.html)
  + Designing tables to meet logical requirements
    - [Cockroach U]
      * [One to Many Relationships](https://www.youtube.com/watch?v=cLceQvauxBU&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=5)
      * [Using JSON Data](https://www.youtube.com/watch?v=ZZVQvIWxFME&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=7)
      * [Computed Columns](https://www.youtube.com/watch?v=LS9Ws9OxypA&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=9)
  + Designing indexes based on query patterns
    - Multi-column, covering, inverted, hash-sharded, data replication and location.
      * [Cockroach U]
        + [Secondary Indexes](https://www.youtube.com/watch?v=zD4AtWZwsEo&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=4)
        + [Covered Queries](https://www.youtube.com/watch?v=LsTZLx32d4Q&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=16&t=0s)
        + [Inverted indexes](https://www.youtube.com/watch?v=nJkuUR3zWIg&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=9)
        + [Composite Indexes and Index Prefixes](https://www.youtube.com/watch?v=bP6bTuIsm6U&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=10&t=0s)
        + [Sorting and Indexes](https://www.youtube.com/watch?v=g-BEXRHZpUQ&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=15&t=0s)
      * [Docs]
        + [Indexes](https://www.cockroachlabs.com/docs/v20.1/indexes.html)
        + [Inverted Indexes](https://www.cockroachlabs.com/docs/v20.1/inverted-indexes.html)
        + [Indexes best practices](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#indexes-best-practices)
  + Designing tables and indexes to meet physical requirements for performance and compliance
    - Key distribution, interleaving, partitioning, duplicate indexes
      * [Cockroach U] [Physical Explain Plans](https://www.youtube.com/watch?v=kFsnBrwQ1nA&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=3)
      * [Docs]
        + [Unique IDs](https://www.cockroachlabs.com/docs/v20.1/sql-faqs.html#what-are-the-differences-between-uuid-sequences-and-unique_rowid)
        + [Column families](https://www.cockroachlabs.com/docs/v20.1/column-families.html)
        + [Interleaving](https://www.cockroachlabs.com/docs/v20.1/interleave-in-parent.html)
        + [Topology patterns overview](https://www.cockroachlabs.com/docs/v20.1/topology-patterns.html)
        + [Configure replication zones](https://www.cockroachlabs.com/docs/v20.1/configure-replication-zones.html)
  + Using a third-party database IDE and database GUI
    - [Docs]
      * [Third-party database tools](https://www.cockroachlabs.com/docs/v20.1/third-party-database-tools.html)
      * [Intellij IDEA](https://www.cockroachlabs.com/docs/v20.1/intellij-idea.html)
      * [DBeaver](https://www.cockroachlabs.com/docs/v20.1/dbeaver.html)
  + How schema changes work in CockroachDB
    - [Cockroach U] [Updating Your Schema](https://www.youtube.com/watch?v=lwg_YRkoBqE&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=7)
    - [Docs]
      * [Online schema changes](https://www.cockroachlabs.com/docs/stable/online-schema-changes.html)
      * [Schema change limitations](https://www.cockroachlabs.com/docs/stable/online-schema-changes.html#limitations)
    - [Blog] [How online schema changes are possible in CockroachDB](https://www.cockroachlabs.com/blog/how-online-schema-changes-are-possible-in-cockroachdb/)
  + Changing the schema of tables/indexes/etc.
    - [Cockroach U]
      * [Computed Columns](https://www.youtube.com/watch?v=LS9Ws9OxypA&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=9)
      * [Using JSON Data](https://www.youtube.com/watch?v=ZZVQvIWxFME&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=7)
      * [One to Many Relationships](https://www.youtube.com/watch?v=cLceQvauxBU&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=5)
    - [Docs]
      * [ADD COLUMN](https://www.cockroachlabs.com/docs/stable/add-column.html)
      * [ALTER COLUMN](https://www.cockroachlabs.com/docs/stable/alter-column.html)
      * [ALTER TYPE](https://www.cockroachlabs.com/docs/stable/alter-type.html)
      * [RENAME COLUMN](https://www.cockroachlabs.com/docs/stable/rename-column.html)
      * [DROP COLUMN](https://www.cockroachlabs.com/docs/stable/drop-column.html)
      * [ADD CONSTRAINT](https://www.cockroachlabs.com/docs/stable/add-constraint.html)
      * [ALTER PRIMARY KEY](https://www.cockroachlabs.com/docs/stable/alter-primary-key.html)
      * [RENAME CONSTRAINT](https://www.cockroachlabs.com/docs/stable/rename-constraint.html)
      * [DROP CONSTRAINT](https://www.cockroachlabs.com/docs/stable/drop-constraint.html)
      * [VALIDATE CONSTRAINT](https://www.cockroachlabs.com/docs/stable/validate-constraint.html)
      * [Computed columns](https://www.cockroachlabs.com/docs/stable/column-families.html)
      * [RENAME DATABASE](https://www.cockroachlabs.com/docs/stable/rename-database.html)
      * [DROP DATABASE](https://www.cockroachlabs.com/docs/v20.1/drop-database.html)
      * [RENAME TABLE](https://www.cockroachlabs.com/docs/stable/rename-table.html)
      * [RENAME INDEX](https://www.cockroachlabs.com/docs/stable/rename-index.html)
      * [DROP TABLE](https://www.cockroachlabs.com/docs/v20.1/drop-table.html)
      * [RENAME VIEW](https://www.cockroachlabs.com/docs/stable/alter-view.html)
      * [DROP VIEW](https://www.cockroachlabs.com/docs/v20.1/drop-view.html)
      * [DROP SEQUENCE](https://www.cockroachlabs.com/docs/v20.1/drop-sequence.html)
  + Managing schema changes from the application
    - [Docs] [Flyway](https://www.cockroachlabs.com/docs/v20.1/flyway.html)
* Chapter 5: Migrating Data
  + Bulk importing into new tables
    - [Cockroach U] [Connecting to CockroachCloud and Importing Data](https://www.youtube.com/watch?v=7pg_bR5zJVc&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=2&t=1s)
    - [Docs]
      * [Migration overview](https://www.cockroachlabs.com/docs/stable/migration-overview.html)
      * [Migrate from CSV](https://www.cockroachlabs.com/docs/stable/migrate-from-csv.html)
      * [Migrate for Avro](https://www.cockroachlabs.com/docs/stable/migrate-from-avro.html)
      * [Migrate from Oracle](https://www.cockroachlabs.com/docs/stable/migrate-from-oracle.html)
      * [Migrate for Postgres](https://www.cockroachlabs.com/docs/stable/migrate-from-postgres.html)
      * [Migrate from MySQL](https://www.cockroachlabs.com/docs/stable/migrate-from-mysql.html)
      * [IMPORT](https://www.cockroachlabs.com/docs/stable/import.html)
  + Bulk import into existing tables
    - [Docs] [IMPORT INTO](https://www.cockroachlabs.com/docs/stable/import-into.html)
* Chapter 6: Reading and Writing Data
  + SQL feature support in CockroachDB
    - [Docs]
      * [SQL Feature Support](https://www.cockroachlabs.com/docs/stable/sql-feature-support.html)
      * [PostgreSQL Compatibility](https://www.cockroachlabs.com/docs/stable/postgresql-compatibility.html)
  + How reads and writes work in CockroachDB
    - [Docs] [Reads and Writes Overview](https://www.cockroachlabs.com/docs/stable/architecture/reads-and-writes-overview.html)
  + Querying data
    - Single table
      * [Docs]
        + [Simple SELECT clause](https://www.cockroachlabs.com/docs/v20.1/select-clause.html)
        + [Filtering rows](https://www.cockroachlabs.com/docs/v20.1/select-clause.html#filter-rows)
        + [Ordering results](https://www.cockroachlabs.com/docs/v20.1/query-order.html)
        + [Limiting results](https://www.cockroachlabs.com/docs/v20.1/limit-offset.html)
        + [Paginate through results](https://www.cockroachlabs.com/docs/v20.1/selection-queries.html#paginate-through-limited-results)
        + [Views](https://www.cockroachlabs.com/docs/v20.1/views.html)
    - Multiple tables/queries
      * [Docs]
        + [Join expressions](https://www.cockroachlabs.com/docs/v20.1/joins.html)
        + [Set operations](https://www.cockroachlabs.com/docs/v20.1/selection-queries.html#set-operations)
        + [Subqueries](https://www.cockroachlabs.com/docs/v20.1/subqueries.html)
        + [Common Table Expressions](https://www.cockroachlabs.com/docs/v20.1/common-table-expressions.html)
  + Computing values from data
    - [Cockroach U] [Computed Columns](https://www.youtube.com/watch?v=LS9Ws9OxypA&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=11)
    - [Docs]
      * [Scalar expressions](https://www.cockroachlabs.com/docs/v20.1/scalar-expressions.html)
      * [Functions and operators](https://www.cockroachlabs.com/docs/v20.1/functions-and-operators.html)
      * [Window functions](https://www.cockroachlabs.com/docs/v20.1/window-functions.html)
      * [Aggregate functions](https://www.cockroachlabs.com/docs/v20.1/select-clause.html#aggregate-functions)
  + Inserting data
    - [Docs]
      * [Insert Data](https://www.cockroachlabs.com/docs/stable/insert-data.html)
      * [INSERT](https://www.cockroachlabs.com/docs/stable/insert.html)
      * [UPSERT](https://www.cockroachlabs.com/docs/v20.1/upsert.html)
      * [Multi-row DML best practices](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#multi-row-dml-best-practices)
      * [Bulk insert best practices](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#bulk-insert-best-practices)
  + Updating data
    - [Docs]
      * [Update Data](https://www.cockroachlabs.com/docs/stable/update-data.html)
      * [UPDATE](https://www.cockroachlabs.com/docs/stable/update.html)
      * [Multi-row DML best practices](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#multi-row-dml-best-practices)
      * [Force index selection for update](https://www.cockroachlabs.com/docs/stable/update.html#force-index-selection-for-updates)
  + Deleting data
    - [Docs]
      * [Delete Data](https://www.cockroachlabs.com/docs/v20.1/delete-data.html)
      * [DELETE](https://www.cockroachlabs.com/docs/v20.1/delete.html)
      * [TRUNCATE](https://www.cockroachlabs.com/docs/v20.1/truncate.html)
      * [Bulk deletion best practices](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#bulk-deletion-best-practices)
  + Using transactions
    - Implicit vs. explicit, nested transactions, handling retry errors (contention vs. non-contention-based retries), transaction size, SELECT FOR UPDATE, AS OF SYSTEM TIME, flattening multi-statement transactions into CTEs to use explicit transactions
      * [Docs]
        + [Understanding and avoiding transaction contention](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#understanding-and-avoiding-transaction-contention)
        + [Run Multi-Statement Transactions](https://www.cockroachlabs.com/docs/v20.1/run-multi-statement-transactions.html)
        + [Transactions Overview](https://www.cockroachlabs.com/docs/stable/transactions.html)
        + [Transaction Retries](https://www.cockroachlabs.com/docs/stable/transactions.html#transaction-retries)
        + [Transaction Retry Error Reference](https://www.cockroachlabs.com/docs/stable/transaction-retry-error-reference.html)
        + [Nested transactions](https://www.cockroachlabs.com/docs/stable/transactions.html#nested-transactions)
        + [Common table expressions](https://www.cockroachlabs.com/docs/v20.1/common-table-expressions.html)
        + [SELECT FOR UPDATE](https://www.cockroachlabs.com/docs/v20.1/select-for-update.html)
        + [AS OF SYSTEM TIME](https://www.cockroachlabs.com/docs/v20.1/as-of-system-time.html)
* Chapter 7: Query Optimization
  + Optimizing inefficient table schemas and relationships
    - No current resources
  + Optimizing inefficient queries
    - EXPLAIN variations, query changes, index changes, indexes with “STORING” for aggregate calculations, index hints, join hints, join reordering, canceling queries, tracing (Lightstep, Jaeger).
      * [Cockroach U]
        + [Physical Explain Plans](https://www.youtube.com/watch?v=kFsnBrwQ1nA&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=4)
        + [Covered Queries](https://www.youtube.com/watch?v=LsTZLx32d4Q&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=16&t=0s)
      * [Docs]
        + [Identify slow queries](https://www.cockroachlabs.com/docs/v20.1/query-behavior-troubleshooting.html#identify-slow-queries)
        + [Table scans best practices](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#table-scans-best-practices)
        + [EXPLAIN](https://www.cockroachlabs.com/docs/v20.1/explain.html)
        + [EXPLAIN ANALYZE](https://www.cockroachlabs.com/docs/v20.1/explain-analyze.html)
        + [Indexes](https://www.cockroachlabs.com/docs/v20.1/indexes.html)
        + [CREATE INDEX](https://www.cockroachlabs.com/docs/v20.1/create-index.html)
        + [Index hints](https://www.cockroachlabs.com/docs/v20.1/table-expressions.html#force-index-selection)
        + [Join hints](https://www.cockroachlabs.com/docs/v20.1/cost-based-optimizer.html#join-hints)
        + [Join reordering](https://www.cockroachlabs.com/docs/v20.1/cost-based-optimizer.html#join-reordering)
        + [CANCEL QUERY](https://www.cockroachlabs.com/docs/v20.1/cancel-query.html)
        + [SHOW TRACE](https://www.cockroachlabs.com/docs/v20.1/show-trace.html)
  + Optimizing inefficient inserts, updates, and deletes
    - Multi-row DML, index overhead, topology patterns, tracing (Lightstep, Jaeger), GC adjustments
      * [INSERT/UPDATE statements are slow](https://www.cockroachlabs.com/docs/v20.1/query-behavior-troubleshooting.html#insert-update-statements-are-slow)
      * [Multi-row DML best practices](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#multi-row-dml-best-practices)
      * [Force index selection for update](https://www.cockroachlabs.com/docs/stable/update.html#force-index-selection-for-updates)
      * [Topology patterns](https://www.cockroachlabs.com/docs/v20.1/topology-patterns.html)
      * [SHOW TRACE](https://www.cockroachlabs.com/docs/v20.1/show-trace.html)
  + Optimizing transactions
    - Reducing contention and transaction retries
      * [Docs]
        + [Understanding and avoiding transaction contention](https://www.cockroachlabs.com/docs/v20.1/performance-best-practices-overview.html#understanding-and-avoiding-transaction-contention)
        + [Run Multi-Statement Transactions](https://www.cockroachlabs.com/docs/v20.1/run-multi-statement-transactions.html)
        + [Transactions Overview](https://www.cockroachlabs.com/docs/stable/transactions.html)
        + [Transaction Retries](https://www.cockroachlabs.com/docs/stable/transactions.html#transaction-retries)
        + [Transaction Retry Error Reference](https://www.cockroachlabs.com/docs/stable/transaction-retry-error-reference.html)
        + [Nested transactions](https://www.cockroachlabs.com/docs/stable/transactions.html#nested-transactions)
        + [Common table expressions](https://www.cockroachlabs.com/docs/v20.1/common-table-expressions.html)
        + [SELECT FOR UPDATE](https://www.cockroachlabs.com/docs/v20.1/select-for-update.html)
        + [AS OF SYSTEM TIME](https://www.cockroachlabs.com/docs/v20.1/as-of-system-time.html)
  + Optimizing inefficient SQL generated by an ORM
    - No current resources
  + Configuring application concurrency and connection pooling for efficient use of database resources
    - [Docs]
      * [Connection pooling](https://www.cockroachlabs.com/docs/v20.1/recommended-production-settings.html#connection-pooling)
      * More/better docs over the next months
* Chapter 8: Working with Spatial Data
  + [Docs]
    - [Getting started](https://www.cockroachlabs.com/docs/v20.2/spatial-features#getting-started)
    - [Migrating spatial data](https://www.cockroachlabs.com/docs/v20.2/spatial-features#migrating-spatial-data-into-and-out-of-cockroachdb)
    - [SQL spatial reference](https://www.cockroachlabs.com/docs/v20.2/spatial-features#reference) (indexes, objects, data types, functions)
* Chapter 9: Planning a CockroachDB Deployment
  + Choosing a platform
    - Self-hosted or managed by Cockroach Labs
    - Cloud, on-prem, or hybrid
    - Kubernetes or process management, etc.
      * No explicit resources
  + Choosing cluster and data topologies
    - For resiliency
    - For latency (multi-region only)
    - For compliance (multi-region only)
    - Anti-patterns
      * [Docs]
        + [Basic topology recommendations](https://www.cockroachlabs.com/docs/v20.1/recommended-production-settings.html#topology)
        + [Topology patterns overview](https://www.cockroachlabs.com/docs/v20.1/topology-patterns.html)

[Basic production](https://www.cockroachlabs.com/docs/stable/topology-basic-production.html)

[Geo-partitioned replicas](https://www.cockroachlabs.com/docs/stable/topology-geo-partitioned-replicas.html)

[Geo-partitioned leaseholders](https://www.cockroachlabs.com/docs/stable/topology-geo-partitioned-leaseholders.html)

[Duplicate indexes](https://www.cockroachlabs.com/docs/stable/topology-duplicate-indexes.html)

[Follower reads](https://www.cockroachlabs.com/docs/stable/topology-follower-reads.html)

[Follow-the-workload](https://www.cockroachlabs.com/docs/stable/topology-follow-the-workload.html)

* + - * + [Configure replication zones](https://www.cockroachlabs.com/docs/v20.1/configure-replication-zones.html)
        + [Topology anti-patterns](https://www.cockroachlabs.com/docs/v20.1/topology-patterns.html#anti-patterns)
        + More explicit docs guidance maybe coming soon
  + Sizing hardware
    - For workload (tps, transaction balance)
    - For storage (data size, data growth)
    - Anti-patterns
      * [Docs]
        + [CockroachDB performance](https://www.cockroachlabs.com/docs/v20.1/performance.html)
        + [Basic hardware recommendations](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#basic-hardware-recommendations)
        + [Cloud-specific recommendations](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#cloud-specific-recommendations)
        + More explicit docs guidance maybe coming soon
  + Defining a load balancing strategy
    - [Docs] [Load balancing](https://www.cockroachlabs.com/docs/v20.1/recommended-production-settings.html#load-balancing)
* Chapter 10: Deploying a Single-Region Cluster
  + Configuring hardware for nodes
    - [Docs]
      * [Clock synchronization](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#clock-synchronization)
      * [Cache and SQL memory size](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#cache-and-sql-memory-size)
      * [Dependencies](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#dependencies)
      * [File descriptor limits](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#file-descriptors-limit)
      * [cockroach debug ballast](https://www.cockroachlabs.com/docs/v20.1/cockroach-debug-ballast.html)
  + Configuring networking for cluster
    - [Docs]
      * [Networking flags and combinations](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#networking-flags)
      * [Single network](https://www.cockroachlabs.com/docs/v20.1/recommended-production-settings.html#cluster-on-a-single-network)
      * [Multiple networks](https://www.cockroachlabs.com/docs/v20.1/recommended-production-settings.html#cluster-spanning-multiple-networks)
  + Starting cluster
    - [Docs]
      * [On-prem](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-premises.html)
      * [GCP](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-google-cloud-platform.html)
      * [AWS](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-aws.html)
      * [Azure](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-microsoft-azure.html)
      * [Digital Ocean](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-digital-ocean.html)
      * [Kubernetes single-region](https://www.cockroachlabs.com/docs/stable/orchestrate-cockroachdb-with-kubernetes.html)
  + Setting up load balancing
    - [Docs]
      * [On-prem](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-premises.html#step-6-set-up-load-balancing)
      * [GCP](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-google-cloud-platform.html#step-4-set-up-load-balancing)
      * [AWS](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-aws.html#step-4-set-up-load-balancing)
      * [Azure](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-microsoft-azure.html#step-4-set-up-load-balancing)
      * [Digital Ocean](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-digital-ocean.html#step-3-set-up-load-balancing)
  + Using replication controls to meet resilience requirements
    - [Docs]
      * [Production Checklist](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#topology)
      * [Configure replication zones](https://www.cockroachlabs.com/docs/v20.1/configure-replication-zones.html)
  + Platform-specific best practices
    - [Docs]
      * [Cloud-specific recommendations](https://www.cockroachlabs.com/docs/stable/recommended-production-settings.html#cloud-specific-recommendations)
      * Cloud tutorials linked above
* Chapter 11: Deploying a Multi-Region Cluster
  + Using data topologies to meet resilience, latency, and compliance requirements
    - [Multi-region topology patterns overview](https://www.cockroachlabs.com/docs/v20.1/topology-patterns.html#multi-region-patterns)
      * [Geo-partitioned replicas](https://www.cockroachlabs.com/docs/stable/topology-geo-partitioned-replicas.html)
      * [Geo-partitioned leaseholders](https://www.cockroachlabs.com/docs/stable/topology-geo-partitioned-leaseholders.html)
      * [Duplicate indexes](https://www.cockroachlabs.com/docs/stable/topology-duplicate-indexes.html)
      * [Follower reads](https://www.cockroachlabs.com/docs/stable/topology-follower-reads.html)
      * [Follow-the-workload](https://www.cockroachlabs.com/docs/stable/topology-follow-the-workload.html)
* Chapter 12: Securing CockroachDB
  + Using authentication features
    - [Docs]
      * [Authentication](https://www.cockroachlabs.com/docs/v20.1/authentication.html)
        + [GSSAPI](https://www.cockroachlabs.com/docs/stable/gssapi_authentication.html)
  + Using encryption features (in flight and at rest)
    - [Docs] [Encryption](https://www.cockroachlabs.com/docs/v20.1/encryption.html)
  + Using authorization features (role and user access controls)
    - [Cockroach U] [Creating Users](https://www.youtube.com/watch?v=XS6iF4SjWoU&list=PL_QaflmEF2e8ijr7gxCZHSKH9-Vl8Yf9q&index=3&t=1s)
    - [Docs]
      * [Authorization](https://www.cockroachlabs.com/docs/v20.1/authorization.html)
      * [SQL users](https://www.cockroachlabs.com/docs/stable/authorization.html#sql-users)
      * [Roles](https://www.cockroachlabs.com/docs/stable/authorization.html#roles)
      * [Privileges](https://www.cockroachlabs.com/docs/stable/authorization.html#privileges)
      * [Access control best practices](https://www.cockroachlabs.com/docs/stable/authorization.html#authorization-best-practices)
      * [Access control SQL statements](https://www.cockroachlabs.com/docs/stable/sql-statements.html#access-management-statements)
  + Audit and maintain cluster security
    - [Docs]
      * [Rotate security certifications](https://www.cockroachlabs.com/docs/v20.1/rotate-certificates.html)
      * [Audit logs](https://www.cockroachlabs.com/docs/stable/sql-audit-logging.html)
      * [Authentication logs](https://www.cockroachlabs.com/docs/stable/query-behavior-troubleshooting.html#authentication-logs)
* Chapter 13: Maintaining CockroachDB
  + Release support policy
    - [Docs] [Release support policy](https://www.cockroachlabs.com/docs/releases/release-support-policy.html)
  + Upgrading clusters with no downtime or performance impact
    - [Docs]
      * [Upgrade a cluster](https://www.cockroachlabs.com/docs/stable/upgrade-cockroach-version.html)
      * [Kubernetes single-region](https://www.cockroachlabs.com/docs/stable/orchestrate-cockroachdb-with-kubernetes.html#upgrade-the-cluster)
      * [Kubernetes multi-region](https://www.cockroachlabs.com/docs/stable/orchestrate-cockroachdb-with-kubernetes-multi-cluster.html#upgrade-the-cluster)
  + Maintaining node hardware with no downtime or performance impact
    - [Docs]
      * [Prepare for planned maintenance](https://www.cockroachlabs.com/docs/v20.1/operational-faqs.html#how-do-i-prepare-for-planned-node-maintenance)
  + Decommissioning nodes with no downtime or performance impact
    - [Docs]
      * [Decommission nodes](https://www.cockroachlabs.com/docs/v20.1/remove-nodes.html)
      * [Kubernetes single-region](https://www.cockroachlabs.com/docs/stable/orchestrate-cockroachdb-with-kubernetes.html#remove-nodes)
  + Scaling clusters for increased capacity, resiliency, or locality (single-to-multi-region)
    - [Docs]
      * [On-prem](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-premises.html#step-9-scale-the-cluster)
      * [GCP](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-google-cloud-platform.html#step-11-scale-the-cluster)
      * [AWS](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-aws.html#step-11-scale-the-cluster)
      * [Azure](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-microsoft-azure.html#step-11-scale-the-cluster)
      * [Digital Ocean](https://www.cockroachlabs.com/docs/stable/deploy-cockroachdb-on-digital-ocean.html#step-11-scale-the-cluster)
      * [Kubernetes single-region](https://www.cockroachlabs.com/docs/stable/orchestrate-cockroachdb-with-kubernetes.html#add-nodes)
      * [Kubernetes multi-region](https://www.cockroachlabs.com/docs/stable/orchestrate-cockroachdb-with-kubernetes-multi-cluster.html#scale-the-cluster)
* Chapter 14: Monitoring CockroachDB
  + Manual monitoring and automated alerting
    - [Docs]
      * [Monitoring tools](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#monitoring-tools)
        + [Admin UI](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#admin-ui)
        + [Prometheus endpoint](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#prometheus-endpoint)
        + [Health endpoints](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#health-endpoints)
        + [Raw status endpoints](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#raw-status-endpoints)
        + [Node status command](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#node-status-command)
      * [Events to alert on](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#events-to-alert-on)
      * [Monitoring with Prometheus, Alertmanager, and Grafana](https://www.cockroachlabs.com/docs/v20.1/monitoring-and-alerting.html#events-to-alert-on)
* Chapter 15: Disaster Recovery
  + Backup and restore options
    - [Docs]
      * [Full and incremental backups](https://www.cockroachlabs.com/docs/v20.2/take-full-and-incremental-backups.html)
      * [Backups with revision history and point-in-time restore](https://www.cockroachlabs.com/docs/v20.2/take-backups-with-revision-history-and-restore-from-a-point-in-time.html)
      * [Encrypted backup and restore](https://www.cockroachlabs.com/docs/v20.2/take-and-restore-encrypted-backups.html)
      * [Locality-aware backup and restore](https://www.cockroachlabs.com/docs/v20.2/take-and-restore-locality-aware-backups.html)
      * [Scheduled backups](https://www.cockroachlabs.com/docs/v20.2/manage-a-backup-schedule.html)
      * SQL reference:
        + [BACKUP](https://www.cockroachlabs.com/docs/v20.1/backup.html)
        + [RESTORE](https://www.cockroachlabs.com/docs/v20.1/restore.html)
  + Recovery from disaster scenarios
    - [Docs]
      * [Disaster recovery](https://www.cockroachlabs.com/docs/v20.2/disaster-recovery.html)
* Chapter 16: Streaming Changes from CockroachDB
  + Intro to Change Data Capture (CDC)
    - [Docs] [Change data capture](https://www.cockroachlabs.com/docs/v20.1/change-data-capture.html)
  + Using Enterprise CDC
    - [Docs]
      * [Stream a changefeed to S3 and Snowflake](https://www.cockroachlabs.com/docs/cockroachcloud/v20.1/stream-changefeed-to-snowflake-aws.html)
      * [CREATE CHANGEFEED](https://www.cockroachlabs.com/docs/v20.1/create-changefeed.html)
  + Using Core CDC
    - [Docs] [CHANGEFEED FOR](https://www.cockroachlabs.com/docs/v20.1/changefeed-for.html)
* Chapter 16: Troubleshooting
  + Support resources
    - [Docs] [Support resources](https://www.cockroachlabs.com/docs/stable/support-resources.html)
  + Reading CockroachDB logs
    - [Docs]
      * [Debug and error logs](https://www.cockroachlabs.com/docs/stable/debug-and-error-logs.html)
      * [Audit logs](https://www.cockroachlabs.com/docs/stable/sql-audit-logging.html)
      * [Cluster-wide execution logs](https://www.cockroachlabs.com/docs/stable/query-behavior-troubleshooting.html#cluster-wide-execution-logs)
      * [Per-node execution logs](https://www.cockroachlabs.com/docs/stable/query-behavior-troubleshooting.html#per-node-execution-logs)
      * [Authentication logs](https://www.cockroachlabs.com/docs/stable/query-behavior-troubleshooting.html#authentication-logs)
      * [Slow query logs](https://www.cockroachlabs.com/docs/stable/query-behavior-troubleshooting.html#slow-query-logs)
  + Cluster/node startup issues
    - [Docs] [Cluster/node startup issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html)
  + Networking issues
    - [Docs] [Networking issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#networking-issues)
  + Cluster configuration issues
    - [Docs] [Clock sync issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#clock-sync-issues)
  + Hardware issues
    - [Docs]
      * [Capacity planning issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#capacity-planning-issues)
      * [Storage issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#storage-issues)
      * [Memory issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#memory-issues)
  + Data replication, data placement, and zone constraint conformance issues
    - [Docs]
      * [Replication reports](https://www.cockroachlabs.com/docs/stable/query-replication-reports.html)
      * [Replication issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#replication-issues)
  + Client connection and authentication issues
    - [Docs]
      * [Client connection issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#client-connection-issues)
      * [Common errors: connection refused](https://www.cockroachlabs.com/docs/stable/common-errors.html#connection-refused)
      * [Authentication issues](https://www.cockroachlabs.com/docs/stable/cluster-setup-troubleshooting.html#cockroachdb-authentication-issues)
      * [Common errors: SSL connection required](https://www.cockroachlabs.com/docs/stable/common-errors.html#node-is-running-secure-mode-ssl-connection-required)
  + SQL feature errors
    - [Docs]
      * [SQL Feature Support](https://www.cockroachlabs.com/docs/stable/sql-feature-support.html)
      * [PostgreSQL Compatibility](https://www.cockroachlabs.com/docs/stable/postgresql-compatibility.html)
* Chapter 18: Cluster Optimization
  + Addressing low throughput
    - [Docs]
      * [Low throughput](https://www.cockroachlabs.com/docs/v20.1/query-behavior-troubleshooting.html#low-throughput)
      * [Hot ranges/nodes](https://www.cockroachlabs.com/docs/v20.1/query-behavior-troubleshooting.html#single-hot-node)
  + Addressing high latency in multi-region clusters
    - [Docs]
      * [Multi-region topologies](https://www.cockroachlabs.com/docs/v20.1/topology-patterns.html#multi-region-patterns)
      * [Low-latency, multi-region deployment](https://www.cockroachlabs.com/docs/stable/demo-low-latency-multi-region-deployment.html)
  + Using archival partitioning to reduce cost
    - [Docs]
      * [Why use archival partitioning](https://www.cockroachlabs.com/docs/v20.1/partitioning.html#why-use-table-partitioning)
      * [Example](https://www.cockroachlabs.com/docs/v20.1/partitioning.html#define-table-partitions-by-range)
  + Managing other cluster operations impacting performance
    - Jobs, GC, automatic statistics
      * [Docs]
        + [SHOW JOBS](https://www.cockroachlabs.com/docs/stable/show-jobs.html)
        + [CANCEL JOBS](https://www.cockroachlabs.com/docs/stable/cancel-job.html)
        + [PAUSE JOBS](https://www.cockroachlabs.com/docs/stable/pause-job.html)
        + [RESUME JOBS](https://www.cockroachlabs.com/docs/stable/resume-job.html)
        + [Admin UI: Jobs Page](https://www.cockroachlabs.com/docs/stable/admin-ui-jobs-page.html)
        + [Disk space usage after deletes](https://www.cockroachlabs.com/docs/v20.1/delete.html#disk-space-usage-after-deletes)
        + [Why are my deletes getting slower oer time?](https://www.cockroachlabs.com/docs/v20.1/sql-faqs.html#why-are-my-deletes-getting-slower-over-time)
        + [gc.ttlseconds setting](https://www.cockroachlabs.com/docs/v20.1/configure-replication-zones.html#replication-zone-variables)
        + [Table statistics](https://www.cockroachlabs.com/docs/stable/cost-based-optimizer.html#table-statistics)
        + [SHOW STATISTICS](https://www.cockroachlabs.com/docs/stable/show-statistics.html)
        + [CREATE STATISTICS](https://www.cockroachlabs.com/docs/stable/create-statistics.html)

## Specs and Schedule

*How many pages do you expect the book to be?*

400-500 pages

*Will you be using illustrations or screenshots?*

Yes, illustrations will likely be used to clarify, for example, the architecture, deployment concepts, and perhaps other concepts. Screenshots should be minimal, but we may need some for the Admin UI and/or related third-party monitoring tools.

*Do any special considerations apply to your plans for the book, including unusual format, use of color, hard-to-get illustrations, or anything else calling for unusual resources?*

No.

*What do you anticipate your delivery schedule to be? If there are any future events/milestones/releases we should consider in our publication schedule, please list those considerations as well.*